LEGISLATING FOR IMPLEMENTATION

CONGRESSIONAL TECH AND PUBLIC POLICY FELLOWSHIP HANDBOOK



Based on the program taught at
Georgetown University's McCourt School of Public Policy.
Fall 2025 & Spring 2026

GEORGETOWN UNIVERSITY
McCourt School of Public Policy

TECH & PUBLIC POLICY

Niskanen Center

Contents

About this publication	1
About the fellowship	1
A Letter to Fellows	2
How to use this handbook	3
Chapter 1: Why policy fails	4
Chapter 2: The Goldilocks problem	10
Chapter 3: Implementation	16
Chapter 4: The Product Operating Model	25
Chapter 5: AI in implementation	29
Chapter 6: Legislating for implementation	34
Appendix A: Key Messages	36
Appendix B: Key Messages	39
Appendix C: Questions to Ask	42
Appendix D: Glossary	45
Appendix E: Reading List	47

About this publication

This handbook is a written encapsulation of the Congressional Tech & Policy Fellowship, and was developed as a companion resource, capturing the key lessons and insights from the program's pilot year.
It was made possible by the support of the Niskanen Center.

It is an attempt to make the invisible visible — to illuminate how legislation is implemented effectively, and to provide insight into how to enable more of what works. While the handbook does not capture everything that was said in the program, it is intended to serve as a reference for fellows, but also for others who may benefit from the knowledge shared.

About the fellowship

The Congressional Tech & Policy Fellowship is a pilot program of the Tech & Public Policy (TPP) Program at Georgetown University's McCourt School of Public Policy, made possible by the generous support of the **William and Flora Hewlett Foundation** and the **David and Lucile Packard Foundation**. The McCourt School gratefully acknowledges both foundations, whose commitment and investments have been instrumental in supporting this work and advancing shared goals.

Designed to be practical and directly relevant to legislative professionals, the program emphasizes that the ultimate success of policy comes through implementation, which is now inseparable from technology. The course focuses on the levers within the reach of legislators to enable better policy implementation.

For years, people working in the civic tech arena have been highlighting the need for public institutions to have the capacity to effectively implement the technology that policy now depends on. McCourt leaned into the opportunity to pilot a program with congressional staff because it is ultimately about something everyone can agree on: *government institutions that are able to deliver on their promises to the public*.

The fellows reflect the breadth of the legislative branch — bipartisan cohorts with representation from both chambers; member offices and committees, varying levels of seniority; and a wide range of policy expertise — united by a commitment to making the legislation Congress passes have the intended impact.

The program was designed and led by Emily Tavoulareas, who brings nearly two decades of experience at the intersection of technology and public service, including as a founder of the U.S. Digital Service and as Senior Advisor to the White House Chief Technology Officer.

A special thanks to

- **Travis Moore** for enthusiastically supporting this effort from day zero and providing guidance on how to make this program useful and relevant to congressional staff
- **Don Moynihan** for advocating for and shaping the foundation of the program
- **Emily Tavoulareas** for designing and leading the implementation of the program
- **Lynn Overmann**, **Katie Ward**, **Aaron Snow**, **Ann Lewis**, **Ben Bain**, and **Amanda Patarino** for providing feedback on the early iterations
- **Marci Harris** for keeping us practically oriented toward the realities of working in Congress
- TPP Deputy Director **Maria Sanchez** and our incredible students **Amr Yakout**, **Chiara Perotti Correa**, **Claire Wooton**, and **Isabel Chiriboga Malo**, who all ensured this program went smoothly from start to finish
- Georgetown leadership for their support, especially TPP Director **Mike Bailey** and **Dean Carole Roan Gresenz**, and Executive Director of Strategic Initiatives **Leslie Evertz**
- And of course all the speakers and panelists who shared their experience and wisdom with us over the pilot year: **Ann Lewis, Marci Harris, Ben Bain, Amanda Patarino, Solitaire Carroll, Mike Wilkening, Eddie Hartwig, Lauren Lombardo, Tara Mcguinness, Aaron Lemon-Strauss, Katie Frost, Marianne Bellotti, Cori Zarek, and Florence Kasule**

A Letter to Fellows

For the better part of 20 years, many of us working at the intersection of technology and public policy have been echoing the same refrain, with varying degrees of success: You cannot separate a policy from the technology it depends on or from implementation more broadly. Doing so has been a root cause of much of the dysfunction that the government has grappled with for decades.

For too long, this argument was treated as hyperbolic and dismissed as "technosolutionist." Then DOGE walked into federal agencies and started taking control of little-known systems, and suddenly everyone could see what those of us who had worked inside these institutions had long understood: Technology is not an extra thing added on to government programs and services; it is the spinal cord of the institution. It can either facilitate or obstruct your goals.

I had the great fortune of helping the Department of Veterans Affairs (VA) build its internal capacity to improve outcomes for veterans, and it was in that work that this belief crystallized. What surprised (and genuinely frustrated) me was how few of my government colleagues understood the degree to which everything they did depended on technology.

By the end of my time in government, I was convinced that one way we could improve how government functions was if the people working in public institutions had a clearer understanding of the role technology plays in the work they do, and what it takes to create an environment in which technology facilitates rather than obstructs their mission.

I designed this program with that conviction (and you) specifically in mind. You are working in one of the most complex environments imaginable, navigating competing incentives that are genuinely difficult for people outside of legislative work to fully comprehend. The pressures are real: CBO scores, lobbyists, coalition dynamics, jurisdictional considerations, and of course, vote counts. All of these forces shape everything that happens in the legislative process, and I do not envy the task of balancing those interests every day.

This fellowship and handbook aim to help ensure that your work has the impact you intend. There's no reason it can't and the proof is already out there. Teams across the U.S. government have been testing and proving a better operational model for over a decade, and it works. From the VA to FAFSA, we have seen what happens when government institutions are given the structure, flexibility, and people who can deliver on their promises. Everything in these pages (and this program) comes from that world of practice.

We are publishing this at a moment when anger, disappointment, and frustration with government are growing, and the question so many are asking is, "Why does government keep failing to deliver?" That question deserves a serious answer, and the serious answer is not that the government is incapable. It is that the gap between passing a law and implementing it has been systematically underestimated and misunderstood, often by the very people with the most power to close it.

We can do better, and we must.

Emily Tavoulareas

Director, Congressional Tech & Policy Fellowship

How to use this handbook

The chapters that follow track the arc and sequence of the program. Each chapter leads with key messages and closes with the same practical orientation: "What does this mean for the people writing the laws? What can legislative staff do to enable better implementation?"

Each chapter can be read independent of the others, although the thesis builds across all six. Since the chapters reflect the weeks of the program, and each week had a slightly different format, the chapters differ slightly in layout, but share the following:

- Key messages
- Common misconceptions
- Questions to ask
- Glossary of key terms
- Reading list

The appendices constitute standalone reference tools.

Chapter 1: Why policy fails

This chapter draws from talks by Ben Bain, the Niskanen Center's Director for State Capacity, and Amanda Patarino, Niskanen's Government Affairs Manager for State Capacity. What follows is a synopsis of their talks and the discussion that followed.

Passing a bill is necessary but not sufficient for successful policy. Success comes through implementation, and implementation depends on technology.

Here, we lay out four key components of policy implementation:

- people
- procedures
- digital infrastructure
- feedback loops

While the digital component is our primary focus in this program, its connection to people, procedures, and feedback loops will be recurring topics that intersect with digital infrastructure in critical ways.

We also focus on what Ann Lewis and Jennifer Pahlka call the "Product Operating Model" that makes modern services work, and introduce how legislation might enable this model for implementers.

Finally, Eddie Hartwig asks us to consider what it means to introduce policy changes into the existing executive branch landscape and encourage policymakers to understand how their new ideas will interact with existing programs.

Key messages

- **Policy cannot be separated from its implementation**. Policy is made or broken by implementation, and implementation now depends in large part on technology.
- **Implementation is shaped by four interdependent components:** people, procedures, feedback loops, and digital infrastructure.
- **Technology is an invisible dependency.** Treating it as an afterthought is one of the most consistent predictors of implementation failure.
- **Good legislation enables good implementation.** Legislation is implemented elsewhere — either by federal agencies or states. The words in a bill are the foundation that implementers build on.

What makes for good implementation?

Implementation is the work of translating policy into outcomes.

This is where most policy lives or dies, and it's shaped by four interdependent components: *people, procedures, digital infrastructure, and feedback loops*.

1/PEOPLE

You know it better than anyone: Public institutions are made up of the people in them, and government programs are created and carried out by people. The skills, incentives, and structures around those people determine what policy can achieve. Hiring constraints, pay scales, budgets, procurement cultures, contracting authority — well-written legislation takes into account the full range of policy constraints and capacities, including the people charged with implementing the law.

It's a myth that government can't attract top technical talent. In practice, experienced, talented professionals at all levels are drawn to the mission and scale of federal service. Recruitment isn't the challenge. The challenge is creating the conditions in which talented, highly motivated people can do meaningful work.

More on those conditions in the Product Operating Model (Ch. 4).

2/ PROCEDURE

The procedural environment in which government operates shapes the things that agencies are permitted to do, how they're expected to do them, and how fast they can do them. This includes procurement rules, compliance requirements, legal reviews, and appropriations structures designed to ensure accountability and responsible implementation of the laws. But the procedures often become the focus rather than the means of implementing the law, and the public often pays the price.

Excessive procedures can also create incentives that fail to reward good outcomes. For example: The CHIPS Program Office needed to hire talent quickly. Remarkably, they took an average of 67 days to bring on talent, besting the government benchmark of 80 days and the government-wide average of 100 days. They ended up exceeding their hiring goal by more than 10 percent. And yet, they were scolded by the Inspector General for not having written a "comprehensive workforce plan." While the report recognized their achievements, the emphasis was on their failure to document their workforce plan.[1]

The tangled message that reports such as these send is: *Yes we want you to accomplish the goal, but what's most important is that you follow strict procedural requirements*.

3/ DIGITAL INFRASTRUCTURE

However you feel about digital technology, it's a critical dependency of public institutions.

When new legislation creates new requirements, those requirements must be translated into changes in real systems, and those systems have constraints, dependencies, and timelines of their own that the legislative process rarely takes into account.

Legislation that does not account for the systems it depends on is legislation that is doomed to fail. That failure might be immediate or long-term, and it might be visible or hidden, but even in a best case scenario, it will fall short of the desired goal.

> Example: The Harry W. Colmery Veterans Educational Assistance Act of 2017, known as the Forever GI Bill, created new eligibility standards for calculating housing stipends. Eligibility standards are essentially an equation, calculated by a computer. Because IT was an afterthought, the role of the system became visible only after implementation efforts began and it stopped working, leaving veterans in precarious situations while the VA updated its systems.

4/ FEEDBACK LOOPS

Effective implementation requires continuous operations, learning, and communication between those writing policy and those implementing it. In practice, this loop is often broken.

Think of any successful project you've worked on. It probably required ongoing collaboration and communication with colleagues. In a decentralized system of government, communication becomes all the more critical

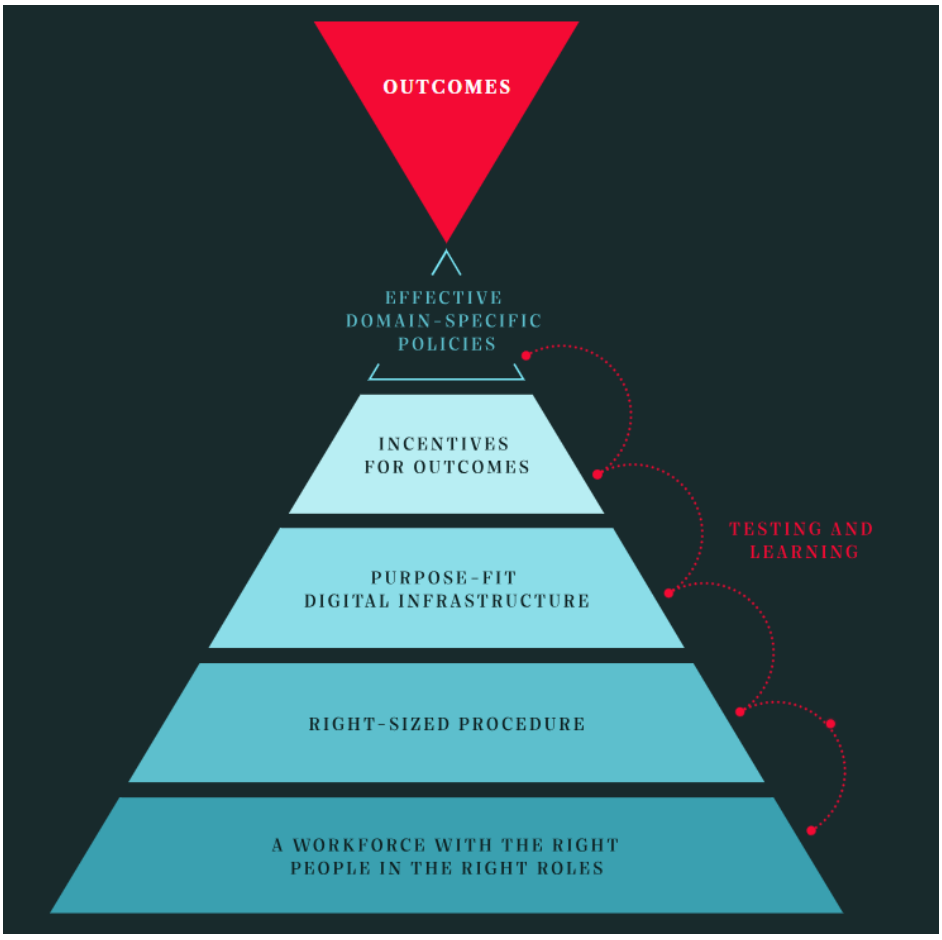
1. Meyers, Sara, "Hiring: Damned If You Do, Damned If You Don't," *Factory Settings*, November 25, 2025.

because policy is written by the legislative branch and implemented by the executive branch or by states. Either way, the people writing policy and those implementing it are entirely different groups of people, with different incentives, constituencies, institutional norms, and modes of operating. This makes an environment of continuous learning and operations even more important. Feedback loops of various kinds enable this.

Civil servants, who have the clearest picture of the realities of implementation, are often either barred from speaking with Congress or have no reliable channel to surface that knowledge to the Hill. And because of the often fear-based relationship between executive branch agencies and Congress, formal legislative liaisons are incentivized to take a defensive posture for the agency rather than to speak openly about issues and to troubleshoot problems.

Hierarchy of state capacity needs

When policymakers create policy, it is to accomplish something. But for the general public, what matters is not a new law or rule or declaration, but rather an outcome that they can see or experience. You can think about it like Maslow's hierarchy of needs, but for good policy outcomes:



Credit: Recoding America [2]

At the bottom of the pyramid you have workforce and procedures — the right people focused on doing the right things. Moving up the pyramid we have "Digital infrastructure," comprising the mission-specific systems and the technologies that enable them; followed by "Incentives for outcomes" encompassing the various feedback loops between policy and implementation; then "Effective domain-specific policies" representing the things we mean

2. "Our Vision," Recoding America, 2026.

when we talk about "policy": energy, housing, transportation, public safety, semiconductors, and so forth. And at the very top are outcomes: how constituents experience government in their daily lives.

As the graphic suggests, we keep churning at the top (outcomes + policy) because we are not addressing the items at the bottom.

How can we improve public policy implementation?

The good news is that there's an operating model that we already know to be effective in carrying out public policy. It's not theoretical or experimental, but is actively practiced in a growing number of federal and state agencies. It's called the Product Operating Model, and we explore it in more detail in Chapter 4.

The basic idea is that rather than treating policy as a project with a fixed specification and a delivery date, the Product Operating Model (POM) treats policy as a live product — one that is continuously tested, improved, and adapted based on what is actually happening in the world. This draws from the world of software development, where there's no "end date" where the product is magically done, but rather an ongoing process of ideation, monitoring, and improvement.

Consider what a product like Amazon looked like when you first signed up, and what it looks like now. There was no end date for a "finished" product, because there was no finish line. The approach was to create something useful and usable, and then monitor how people used it, conduct ethnographic-style research with users, and continuously identify needs, opportunities, and improvements. The only end date is the end of the product.

Ultimately, the Product Operating Model treats policy as an ongoing product that needs continuous improvement, and assumes that policy implementation is never really done — it must adapt to changing needs and opportunities.

Where the Product Operating Model has been successfully employed, the results have been clear. For example: getting the Free Application for Federal Student Aid (FAFSA) back online.

> In December 2023, the Department of Education launched a redesigned FAFSA system that left millions of students unable to access financial aid. The failure followed a now-familiar pattern: vendors working in isolation, a single large deployment, and rigid specifications written without real understanding of user needs. It was the project model at its worst.
>
> The fix came from shifting to the Product Operating Model. A new team took over in 2024 and changed how the work was done: they hired engineers and product managers with real decision-making authority, contracted for team capacity rather than specific deliverables, and let user data drive priorities instead of contract specifications. When data showed a million students getting stuck at a specific step, the team fixed it immediately — even though a different feature had been written into the contract. That kind of decision is only possible when the goal is defined as helping users, not delivering a set of rigid predetermined requirements. By mid-2025, over 14 million students had completed the form, with a 93 percent satisfaction rate.

This kind of implementation requires skills, appropriate procurement structures, and high-functioning relationships between agencies and the Hill that are distinct from the status quo. It also requires an environment that *enables* this way of operating.

Legislation is the foundation on which implementation is built, and it can either enable this model or obstruct or complicate it.

Failure in complex systems

Governments are complex systems. To understand why policies fail, it's important to understand complex systems:

- They are inherently dangerous.
- They are heavily defended against catastrophic failure.
- They are constantly breaking and operating on the edge of catastrophe.
- There is no single cause of catastrophic failure.
- People almost always prevent catastrophe.
- Ex post facto thinking that people should have known better is often wrong.
- All actions introduce both risk and opportunity.
- The longer a system has been deployed, the harder it is to fix.

Small failures don't take down the complex system, but they hinder the system by creating layers of complexity that derail from the original intent of the system.

Similarly, government programs are regularly failing, but catastrophes are rare:

- They provide an opportunity to reduce complexity in the course of addressing the failure (healthcare.gov is a good example).
- But they can also make it worse by triggering overcorrections.
- Real change will not occur if members of the executive branch are not able to communicate honestly. This is challenging because the process requires a willingness to acknowledge problems, which the executive branch is not incentivized to do.

The unsatisfying reality is that there is no single policy solution when operating in a complex system.

Risks appear when changes occur, but change of some kind is necessary when the status quo is no longer tenable. Aversion to risk is partly why government programs operate on ineffective, often dysfunctional systems. On the flip side, disruption that is untethered from reality introduces unproductive and irresponsible risk to the system. Making effective changes requires a clear goal, and a willingness to listen to people doing the work.

> For example: The Supplemental Nutrition Assistance Program (SNAP, or food stamps) was codified into law in 1964. Ten years later, Congress created a similar program, the Special Supplemental Nutrition Program for Women, Infants, and Children (WIC). WIC was written with different rules for eligibility and family size. Now, a family making 190 percent of the poverty line who applies for WIC will be denied, but if they apply for SNAP they will be approved, and automatically enrolled in WIC as well. The confusion families face has prevented WIC programs from digitizing or simplifying the application process, slowing processing times and increasing the cost of administration.

Ideally, policymakers will understand how their new proposal will interact with existing policies, systems, and programs before implementation begins. Doing so will help them adjust their proposals or the existing policies to allow for effective, efficient implementation.

Common misconceptions on implementation

MISCONCEPTION	REALITY
"Technology is an implementation detail, and the agency can figure that out later."	Technology is policy infrastructure. It has dependencies, timelines, and constraints that must be understood before a bill is finalized, not after. Treating it as an afterthought is one of the most reliable predictors of implementation failure.
"If we write it into the bill, agencies will do exactly what we want."	Mandating an outcome does not create the capacity to deliver it. Agencies can only do what their people, systems, and processes allow. Legislation that ignores these constraints sets implementation up for failure.
"The government cannot attract good technical talent."	The mission and scale of public service work are significant draws. Attracting talent is not the main challenge; creating environments in which talented professionals can do meaningful work is.
"This IT product should have been done months ago."	Yes, at some point the product needs to launch and be usable, but software is never really "done." This is both because there is always room for improvement based on actual use and because of how quickly the technology landscape changes. There must be room for ongoing adjustments of all kinds.

Questions to ask

- What technical systems does this bill depend on? Who knows those systems best? Have we found and spoken with them?
- Have we spoken with the agency staff who will implement this — not just formal legislative affairs offices, but the people closest to the work?
- What feedback loops exist between the Hill and implementing agencies for this program? How can we keep an open channel? Have we set the tone and priorities? Have we asked for demos instead of written reports and arbitrary metrics?
- Does the implementing team have the people, budget, procedures, and digital infrastructure to succeed? If not, does the bill address those items?
- What would "good implementation" of this bill look like? How would we know if we'd achieved it? How can we work with the agency to identify indicators of success and failure?

Glossary

Legislative intent: What lawmakers meant when they wrote a bill. Often differs from what the statutory text actually says and from how the executive branch operationalizes it, creating multiple points of potential failure.

Procurement: The government process for contracting with external vendors for goods and services. Federal procurement rules are governed by the Federal Acquisition Regulation and tend to favor large, established vendors and specifications determined before any work is actually done.

Product Operating Model (POM): An iterative, user-centered approach to delivering digital services borrowed from the tech industry in which cross-functional teams continuously test, learn, and improve. Contrasts with the traditional "waterfall" model still prevalent in most government procurement.

Waterfall development: A linear and rigid development process in which each phase (design, build, test, deploy) must be completed before the next begins. Specifications are determined before work begins, leaving no room for learning or adjustments. Common in government software procurement and development.

Chapter 2: The Goldilocks problem

This chapter draws from a talk by Eddie Hartwig, founder of the organizational consultancy Service Design Collective, and discussions among Lauren Lombardo, Policy Director at Recoding America; Aaron-Lemon Strauss, Executive Director of the FAFSA Program; and Katie Frost, former Associate Administrator for Capital Access at the Small Business Administration. What follows is a synopsis of their talks and the discussions that followed.

Two extremes derail policy implementation in different ways: treating technology as an afterthought, and over-prescribing technical specifications that lock in failure.

Here, we explore how to provide clarity while maintaining flexibility for feasible policy implementation at both the state and federal levels.

Key messages

- **There are gaps between what a law says, what it intends, and how it gets operationalized.** Each can have different purposes and goals.
- **Anything left unaddressed in the legislative process can end up in court.** Clarity facilitates implementation.
- **Legislation needs to be clear, but not necessarily specific.** Over-specification creates its own failures.
- **The people who know what's broken often cannot reach the people who need to know.** This leads to overly sanitized information and defensive rather than substantive and constructive conversation.

The "Three Bears" of implementation

Good legislation finds the right balance among three competing pressures:

1. The principles behind a policy
2. The outcomes it is trying to achieve
3. The compromises required to pass it

Those compromises shape what the legislation actually says, which then influences what the executive branch can and will implement and its prospects in litigation. Legislation that is too vague creates operational complexity and invites litigation and judicial review. Legislation that is too specific creates rigidity that makes effective implementation nearly impossible.

The goal should be clarity, not specificity.

While legislative professionals have little to no control over policy implementation itself, they are in a position to incentivize environments that are flexible and adaptable.

The gap between intent and reality

There are at least three versions of every law:

1. What lawmakers intended
2. What the text actually says
3. How the executive branch operationalizes it

The law is often different from the way the statute is understood. Gaps exist between legislative intent and what the law actually says on paper, and between what the law says on paper and how it gets operationalized.

Each transition is a point of potential failure. Agency leadership and lawyers are the people who must translate statutory text into actionable rules, and their ability to act is constrained by what the law actually says, not what policymakers meant or what others claim they meant.

Computers make this problem more concrete, because they can only be programmed around what can be defined precisely. Policy implementers commonly call these "bright-line tests," and they are translated into business requirements for technical teams to program into software. This often works best with "yes" or "no" questions or numerically driven decisions. Determining "better" or "worse" is not so simple. Eligibility criteria that seem reasonable on paper can be nearly impossible to translate into code.

Hard deadlines and the illusion of accountability

Hard deadlines feel like accountability, but in practice, they rarely are.

For example: The 21st Century IDEA Act required every government form to be digitized within two years. There are roughly 34,000 federal forms. Federal CIOs immediately used their time to make the case that the deadline was unachievable.

A more effective approach is the use of progressive deadlines. Keeping with the digitized forms example, a progressive deadline would mean requiring each agency to digitize 10 percent of its forms per year. This is achievable, creates momentum, and produces a useful side effect: Agencies discover forms that are outdated, duplicative, or no longer needed.

Bottom line: Hard deadlines may signal urgency, but they rarely create it. They can also create perverse incentives such as those illustrated by the 2014 scandal at the Department of Veterans Affairs, in which frontline VA workers falsified client-appointment data to meet productivity targets because the well-intentioned goal had been defined as hitting an arbitrary metric, instead of addressing the obstacles to accessing care.

It is better to aim for concrete and measurable outcomes, while providing room for agencies to determine how to achieve them.

Feedback loops

Effective implementation requires continuous communication between those writing policy and those delivering it. In many places, that loop is broken.

Civil servants closest to the work — the ones who can see what is not working and why — often have no reliable way to surface that knowledge to the Hill. Hierarchy, institutional risk aversion, and the fear of personal or institutional consequences all suppress the upward flow of information.

Again: Not everything requires a bill; there are less formal tools that help.

DEMOS
Requesting a demonstration of software designed to deliver a policy outcome is an effective way both to see progress toward a policy goal and to build a relationship with civil servants working directly toward those goals. Push for a real, live demonstration in a test environment, not a slide deck of screenshots or a previously recorded video. Unlike a report, a demonstration also fosters a dialogue between the executive and legislative branches. When live demonstrations are not available, wireframes and user research data early in the project timeline can provide useful topics for discussion.

LETTERS
A letter from Congress to an agency is a legitimate and underused instrument — especially a private letter. If an agency doesn't respond, escalating by copying OMB removes the public shame while retaining the pressure. Letters don't have to be adversarial; they can function as collaborative oversight. Retaining an adversarial, public

letter as a last resort also incentivizes agencies to engage. Letters can also be sent to highlight positive impact and help send the message that good work should continue.

STAFF BRIEFINGS
Informal and regular staff briefings are commonplace and a well-known way for Congress to track executive branch progress. Developing a regular cadence that makes sense within the product development cycle allows Congress to see progress over time and develop deeper relationships with the agency and staff delivering on the policy objectives.

INFORMAL RELATIONSHIPS
The most valuable information often comes through informal conversation. Building those relationships before a crisis is one of the most valuable investments a legislative office can make. These kinds of relationships can help you impress on implementers the intent of the policy and illuminate key questions:

- What is the question I should ask?
- Are we looking for the solution in the right place?
- Are we starting this effort facing the right direction?
- Does the capacity (e.g., staff, budget, authority) exist to succeed?
- What are the primary barriers to success?

> For example: Other transaction authority (OTA) was a new procurement tool that Congress created to increase the use of innovative products from new companies, largely aimed at NASA and the military. In its initial years, authorized agencies did not use this new and powerful capability because they 1) did not fully understand the intent and 2) had not developed ways to integrate it into their procurement skill set. Over several years, informal feedback from executive branch staff, coupled with strategic changes incorporated into subsequent defense authorizations, led to wider adoption of OTA and policy refinements that made it more effective in practice.

There is no perfect policy. Even a perfectly implemented policy, which has never existed, would still be improvable. The goal is better, not perfect. That requires building in mechanisms for correction, not just compliance, and those start with the ability to monitor and respond to progress and issues.

Engaging expertise

Given these broken feedback loops, Congress often leans on outside expertise for a variety of needs. But those outside parties don't always understand (or prioritize) government-specific constraints, and the resulting legislation reflects that gap. The results can be costly financially and politically, and ultimately fail to have the intended impact.

What's the best way, then, to bring in expertise without compromising the policy?

- **Request demos of products or systems.** This allows you to invite more technical staff who are closest to the work and will lead to a more relevant and substantive conversation.
- **Ask staff who are closest to the work to attend briefings.** This will enable a more honest and nuanced conversation about the complexities of implementation and create the opportunity to address potential issues before they arise.
- **Request input and feedback from key stakeholders before decisions are made.** This signals an interest in understanding the realities of their workflow, and can build trust. It also provides an opportunity to understand the limitations of systems, and if and where efficiencies can be achieved.

- **Engage when the stakes are lower.** High-pressure situations can lead to defensive or risk-averse postures and make clarity harder to achieve. Engage when things are going fine, ask questions, and be curious about how things work in practice: What works well? What doesn't? Why?

Risk and the 'Legislative Anxiety Cycle'

"The law tends toward binary interpretation, but risk is a spectrum." – Eddie Hartwig

Government cannot operate effectively at the extremes — neither with the risk tolerance of a venture capital firm nor with the paralysis that comes from treating every decision as a potential catastrophe. The stakes of each program should determine where on that spectrum an agency operates. Because a nuclear weapons program and a pilot initiative for digitizing government forms do not pose the same kinds or levels of risk, they should not be governed the same way.

Each agency draws lines related to risk in different places and for different reasons. Thinking through where those lines should be, and why, requires nuance, not binary thinking.

Fear often drives overprescription, micromanagement, and a bad outcome. A bad outcome then leads to a more specific mandate. That specific mandate inevitably produces new failures. Niskanen calls this the "Legislative Anxiety Cycle."



The Legislative Anxiety Cycle, The How We Need Now, Niskanen Center [3]

The cycle is a downward spiral. Effective oversight requires trust. When trust is absent, legislation often becomes more prescriptive. When mandates become more prescriptive, agencies have less flexibility to deliver. When delivery suffers, trust erodes further.

> Katie Frost, former Associate Administrator for Capital Access at the SBA, shared her experience of the Legislative Anxiety Cycle: During the COVID-19 pandemic, SBA rolled out several programs to support businesses and local economies. Fraud and misuse in a few of these early programs (PPP and COVID EIDL) led to a massive overcorrection in the next program (SVOG). The SVOG team became so afraid of approving anything that in six weeks it approved only about 100 of 14,000 applications (by comparison, a similar program called RRF funded 16,000 applications in one week).

3. Jennifer Pahlka, and Andrew Greenway, *The How We Need Now: A Capacity Agenda for 2025 and Beyond*, Niskanen Center, December 2024.

We already know what works, because we've seen it in pockets across public institutions: better relationships, clearer outcome definitions, a genuine interest in understanding and solving problems, and the willingness to talk to the people closest to the work. We need more of what works, not the path of least resistance or the most visible disruption.

Common misconceptions

Below are some common statements or beliefs related to legislation and accountability.

MISCONCEPTION	REALITY
"Writing it into the bill will force the agency to do it."	Mandating an outcome doesn't create the capacity to deliver it. Overspecification creates rigidity and, often, new failures.
"Hard deadlines create accountability."	They signal urgency but rarely produce it. They can also produce perverse incentives when the goal becomes achieving an arbitrary metric, rather than achieving the intended outcome.
"The law says what we meant."	Legislative intent, statutory text, and agency operationalization are three different things. Assume gaps among all three.
"If experts say it's a good idea, we can write it into law."	Outside experts may not understand government-specific constraints. The bill will reflect whatever knowledge was — or wasn't — in the room.

Questions to ask

- What are we trying to achieve here?
- What technical systems does this bill depend on? Who knows those systems best? Have you talked with them?
- Can a computer evaluate the eligibility criteria in this bill? Think: bright-line tests. If not, how will it be implemented?
- Are the deadlines in this bill achievable? What happens if they aren't met?
- Who will be leading the implementation? Have they been involved from the beginning?
- How will we know if the law being implemented is working? What are indicators of success? What are indicators of failure?
- How can people who will implement the law surface challenges? What channels can we set up to increase feedback?
- What could go wrong? If something does go wrong, as it will, how would we know? How would we respond?

Glossary

Conway's Law: A principle from software engineering stating that systems tend to mirror the communication structures of the organizations that build them. In government, fragmented agencies produce fragmented technology.

Legislative Anxiety Cycle: A downward cycle in which fear and anxiety lead to legislating specific instructions – more mandates, process, and procedure. This over-specification backfires, and weakens agencies' sense of responsibility and accountability to deliver outcomes.

Legislative intent: What lawmakers meant when they wrote a bill. Often differs from what the statutory text actually says, and differs again from how the executive branch operationalizes it, creating multiple points of potential failure.

Oversight: Congressional monitoring of executive branch implementation. Can be hearings, letters, reporting requirements, or informal communication. Most effective when continuous and collaborative rather than reactive.

Progressive deadlines: A sequenced series of achievable milestones that compound over time, in place of a single hard deadline. Designed to create momentum and measurable progress by creating visible cumulative outcomes, rather than a single binary pass/fail at an arbitrary date.

Chapter 3: Implementation

This chapter draws from talks by Solitaire Carroll, former Acting Director of the Veteran Experience Services Portfolio at the Department of Veterans Affairs, and Mike Wilkening, former Secretary of the California Health and Human Services Agency. What follows is a synopsis, in a question-and-answer format, of their presentations and subsequent discussions.

Legislation is not implemented by the same people who write and pass it. Implementation is done by people in executive branch agencies, or in the states.

At the federal level, members of Congress and congressional staff rarely have a view into what happens after legislation is passed, enacted, and implemented: How do the words of a law become a reality? Who implements the law and its different provisions? How? What interactions do those who write the laws have with those who implement the laws? How can these processes be improved?

Here's what we learned.

Key messages

- **Passing a bill is the beginning, not the end.** Implementation happens at the state level or in executive branch agencies. When Congress sets out to write and pass a law, the federal government and states need to figure out how to implement the law should it go into effect.
- **Federalism multiplies the complexities of implementing laws.** Because many federal programs are administered at the state level, there is significant state-to-state variation in how states interpret the law and in the capacity and infrastructure that they can bring to bear.
- **Procurement is policy.** The rules governing how agencies acquire technology and services shape what gets built and how well it works, often to a greater extent than the legislation itself.
- **What Congress does before a bill passes matters enormously.** Early funding, flexible appropriations, general IT language, and robust technical assistance are some of the most valuable resources and activities that the legislative branch can provide.
- **You cannot just bolt new requirements onto legacy systems.** There is no good track record of building new functionality on top of old infrastructure. The assumption that existing systems can absorb new mandates is one of the most reliable predictors of failure.
- **Budget lines usually do not correspond to outcomes.** Development, infrastructure, and platform costs are distributed across multiple systems and funding sources. Until budget structures reflect how implementation actually works, the basic questions — how much was spent, what was the return — will remain difficult to answer.
- **Multiyear funding changes behavior.** Single-year funding creates pressure to spend whatever is left at year's end, on anything. Five-year tech windows allow agencies to plan effectively and make responsible decisions rather than rushed ones.
- **The product model works; the project model doesn't.** Sixteen months figuring out requirements produces solutions that are already outdated when they ship. Start with user needs, work iteratively, and define success through outcomes, not compliance with a plan.

State Level: Implementing the Affordable Care Act

The following is a synopsis of our conversation with Mike Wilkening, former Secretary of the California Health and Human Services Agency, who helped lead the state's implementation of the Affordable Care Act (ACA).

When Congress passes a law and the president signs it, what happens next depends entirely on who is responsible for implementation, and in many cases, that's the states. The ACA rollout was a multiyear process at the

state level, on the policy side alone. States had to decide whether to expand Medicaid, and whether to build their own exchanges or use the federal one.

California decided to do both — build its own exchange and expand Medicaid — while also navigating a gubernatorial transition from Arnold Schwarzenegger to Jerry Brown.

What happens after a bill becomes a law?

At the state level, a federal bill becoming law is just the start. The immediate questions are:

- What are we required to do?
- Are there optional components? If so, do we want to opt in?
- What do we need to do to implement this?
- How much will it cost?
- Where does the money come from?
- Who do we need hire?
- Who are our implementation partners?

At each of those steps, people at agencies, in the governor's office, and elsewhere are continuously discussing and debating what will be needed. These interactions happen informally and formally through the state's legislative process.

California's ACA rollout is a good example. The state decided to build its own exchange and expand Medicaid, while navigating a gubernatorial transition mid-process. California passed a series of bills in 2010 to enable the implementation of the ACA. Governor Schwarzenegger signed these bills in the last year of his governorship. In designing these bills and the structure for implementing the ACA, California had a leg up: Schwarzenegger had already attempted healthcare expansion at the state level, so existing relationships and some familiarity with the levers of change were already in place. Despite a budget crisis at the beginning of his term in 2011, Governor Brown decided to reaffirm California's decision to expand Medicaid and establish Covered California, the state's health insurance exchange.

Over the next couple of years, California passed a series of additional bills necessary to effectuate the implementation of the ACA and the expansion of Medicaid. There were numerous policy debates leading up to the passage of each of these bills. Additionally, there were accompanying budget debates over implementing the eligibility and enrollment system, hiring state and county staff to administer the new and expanded programs, and to issuing grants to partners to assist in signing people up. Aside from the policy and budget processes, the state and counties also had to manage processes for contracting, hiring, and oversight.

How did you manage friction with state oversight bodies?

When you are looking at a huge project, there are only so many things you can change: deadline, scale, and cost. For the ACA, including the Medicaid expansion, implementation was fixed on January 1, 2014. Often, just complying with the internal approvals and checkpoints for a major project leads to years of implementation.

CalHEERS, California's system for the ACA, was exempted from many state rules and regulations. This created friction with the California Department of Technology (CDT) because our implementation wasn't going to fall under its normal oversight structure. Our response was to make clear that its presence was still needed, but that we just couldn't abide by its standard multiyear cycle given the urgency of the mandate. We communicated that directly and kept the department involved.

We understood the intent of the rules, and we found a way to meet that intent without disrupting the process.

How did you move faster than normal state processes allowed?

The Department of Technology has a standard process that includes the budget process along with CDT's own approval processes and oversight. We still issued an RFP, but we stripped down the approval and oversight processes leading to issuance of the contract and delivery of the system. What normally would have been a multiyear effort was completed in 14 months. We also provided high-level engagement throughout the process. Decisions that required cabinet-level or departmental leadership were made in hours or days rather than weeks or months.

You can't buy time back, but you can fix a wrong decision.

In hindsight, what would you have done differently?

I would be more deliberate about what we could safely strip from the normal process. Knowing what you can throw out (and what you can't) is itself a skill, and one worth developing before you're in the middle of a fast-moving implementation.

What does the federal government misunderstand about states in this context?

When Congress passes a bill, that's just the start. The federal government is figuring out implementation at the same time the states are, and if there are budgetary or policy decisions, these things take time. When a bill is the cornerstone of an executive's domestic agenda, things move differently. When a governor gets involved, agencies and officials are pushed to do things they normally wouldn't, or would do much more slowly. When observers see things moving faster in some circumstances than others, what they're often seeing is the stripping of normal processes in order to deliver.

Also, you cannot go into legacy systems and try to build out new functionality for large projects. It will fail. Some innovation is required, and if it sticks, great. But there is no good track record of bolting new requirements onto old systems.

Did moving that quickly mean sacrificing accountability and transparency?

Yes. We were not as transparent as we normally would have been. I knew what was happening, but it wasn't going through the normal checks. We had all the right departments and people involved, but all of us were pushed outside our normal comfort zones. Public engagement and even some traditional stakeholder engagement were limited. These decisions were weighed against the implementation deadlines and consequences of slowing down. There is real value in transparency, and it was a genuine tradeoff, not something to dismiss.

Agency level: Implementing the PACT Act

The following is a synopsis of our discussion with Solitaire Carroll, former Acting Director of the Veteran Experience Services Portfolio at the Department of Veterans Affairs. Carroll helped lead implementation of the Sergeant First Class (SFC) Heath Robinson Honoring our Promise to Address Comprehensive Toxics (PACT) Act at the VA.

What did the PACT Act actually change, and what did implementation require?

The PACT Act was a massive recalculation of benefits eligibility for veterans. It didn't add new programs, but it changed who could access existing ones. Specifically, it removed the requirement for veterans to prove a connection between their service and their condition.

In practice, this meant a need to edit the eligibility criteria, almost doubling the number of veterans who were eligible. This was great, but it introduced huge implementation challenges and risks. Doing things as they had been done with similar efforts in the past would have led to failure.

The implementation timeline ran from the legislative process through four phases:

1. Pre-launch
2. Initial rollout
3. Scaling and optimization
4. Sustained operations with continuous improvement

The PACT Act was so big that everyone had to work together across different VA administrations and workstreams, in fully cross-functional teams that included IT, HR, outreach, and program officers, among others.

The legislative process was still ongoing in August 2022. Why did you start thinking about implementation in May?

We knew this would be complex, and we wanted to get out ahead of it.

We started with cohort analysis: how many veterans would this impact? We estimated around 6 million. Keep in mind the VA processes 5 million claims per year, so if people were to apply right away after enactment of the PACT Act, we would be doubling the backlog overnight. So before the law even passed, we asked for money to clear the existing backlog and finish migration work already underway. We needed to start with a clean slate and no tech debt. The system could not crash underneath the weight of these additions, so that was the first priority.

There was also significant planning required for the rollout phase: hiring more nurses, doctors, and claims processors, and building continuous predictive analytics and feedback loops to fix issues as they emerged.

What did the legislative side not account for in the PACT Act implementation?

Timelines. Not initially. Within the law there were many deadlines to set up reports and create data exchanges between agencies that proved difficult to meet.

The IT throughputs and legacy systems were also underestimated. People didn't realize that VBMS (the Veterans Benefits Management System) on its own couldn't do what was being asked of it. There had to be a lot of education on the limitations of the system. There was also not enough focus on end-to-end service delivery bottlenecks.

How should policymakers best be educated on implementation bottlenecks?

A lot of times a law comes out, someone hires somebody, and they regurgitate things that have already been done. What worked with the PACT Act was that the bill was written and developed by full-time employees who understood the system from the inside.

From there, we had regular, topical deep dives with Congress — casual and brief, with ongoing engagement and feedback with congressional offices — to build understanding and trust over time. With the PACT Act funding, Congress knew exactly where the money was going. That transparency matters.

On the technical side, automation helped significantly. Rather than having a person read 200 pages to find relevant evidence, the team built a language scanning tool to do it instead.

Should we work with appropriators on one-year or multiyear funding?

Multiyear funding is the best kind. With a continuing resolution, you don't know what you're actually getting until the budget passes, and most money ends up spent in just a handful of months of the fiscal year as a result.

With single-year funding, people spend whatever is left at the end of the year on anything, just to avoid losing it. The behavior is better under multiyear funding: If you don't spend it, you get it back next year, so you make the most fiscally responsible decision rather than rushing to obligate funds.

Five-year tech windows have the most impact. They encourage people to do the work effectively and allow for real multiyear transition planning.

How did you get appropriators comfortable with the scale of investment required?

The VA was already underfunded, so the first step was showing appropriators the gap: what the outcomes were, what it actually cost to deliver them, and what modernization required. We drafted a detailed modernization plan: 70 pages of specific, measurable initiatives. Then we reported regularly on progress against those initiatives. The formula was a clear vision and consistent transparency.

What do large government-technology projects have in common when they fail?

They attempt to solve too many complex problems at once. They deliver outdated solutions. They are vendor led, with agencies in a limited role. Expensive and wasted resources is usually the norm.

The underlying issue is the project model, which is the way government typically operates. People spend sixteen months figuring out requirements, which leads to wasted time and money, and by the time something ships, it's already outdated. Working off of rigid, pre-determined requirements in a linear manner sets you up for failure.

The alternative is the product model: start with user needs, begin targeted work immediately, test with users and stakeholders, and deliver regularly. This creates cumulative outcomes that allow you to assess if you're on the right track or not, and if not, you can adjust. This is how highly functional parts of the private sector operate: cross-functional delivery teams, outcome-oriented leadership, modular contracts, and a workforce built to iterate.

On the legislative side, the same logic applies:

- Define problems to solve instead of prescribing solutions
- Focus on outcomes, not features
- Encourage pilots and iteration
- Define success through evidence and observable impact
- Engage with agencies in ways that empower delivery rather than constrain it

Why don't congressional investments align cleanly with outcomes?

Budget lines align to a program office, product name, or acquisition, but that does not reflect the realities of implementation. Rarely does one budget line correspond to one outcome, because of the cross-sectional extent of most IT programs. Development costs, infrastructure costs, and platform costs are all distributed across multiple systems. A single congressional investment can have money flowing to all three, with different contracts under each system paid for by different funding sources. This makes it nearly impossible to answer the basic questions: How much was spent? What was the return?

This is also a feedback-loop problem. Any time an agency is asking Congress for more money, it has to write a congressional budget justification that aligns with some project, but the money is actually used across different line items. Communication back and forth between an agency and Congress on this is a huge effort, especially if the agency wants to change how the budget looks. One of the first myths I heard at the VA was that Congress likes the structure of the budget and doesn't want you to change it.

What makes the timing so difficult?

There is a fundamental mismatch between the timelines of appropriation and actual budgeting. Contracts for a single system are executed multiple times throughout the year, and any change to the execution of the budget or the availability of funds puts the acquisition timeline in jeopardy. Continuing resolutions don't function as the stopgap measure they are intended to be. Carryovers are frowned upon and hard to ask for.

The reality is there is always a reason to spend money; you never run out of technology to invest in. The question is whether procurement and acquisition lifecycles are limiting your ability to spend on what is actually most important at the time you need it.

What is the alternative?

I call it "capability-based budgeting." Group everything related together, so that when extra money comes in you can point at a bucket and make prioritization choices on the fly. Budget utilities together — things that belong together and depend on each other. Don't group utilities with streaming services.

The mindset is this: I have been given this money to achieve these things. I need to be a good steward of it, so I need to spend it on the priorities I grouped together. Regardless of political changes, administration changes, or congressional requests, the buckets themselves shouldn't change. The order in which you prioritize them can change, but the content of each bucket should not.

When you do this well, you are also able to point at an outcome associated with each bucket. That is what makes it legible to Congress, and what makes it possible to have a real conversation about results.

Lessons Learned: The PACT Act

This is directly drawn from Solitaire Carroll's presentation.

WHAT THE LEGISLATIVE BRANCH DID WELL	Early funding with flexible appropriations. Congress provided the VA with upfront resources before the expanded benefits went into effect. This allowed the VA and its teams to prepare for the enormous upcoming effort and to address existing technical debt. Critically, these resources were not for the expanded benefits themselves; they were for the work needed to make expansion possible: the hiring, the technical modernization, and the systems preparation. Allowed general IT language. The PACT Act's modernization language was not overly prescriptive. Congress did not dictate specific features or technical solutions, which gave the VA room to make the right implementation decisions for its systems. Accepted VA technical assistance and adjusted the bill before passage. Congress engaged in extensive back and forth with the VA on timelines, presumptive conditions, and systems impacts, and was willing to adjust the bill based on what they heard. This kind of prepassage collaboration is rare and made a measurable difference. For example, it was originally planned to be a staggered rollout, but the VA objected because analysis showed this would be an issue and would delay the total time for rollout. Congress listened! One important note on this: When legislation passes with funding attached, Congress needs to know what it's going to get for that money. But the executive branch will more often than not do exactly what is written down, and not think outside the scope of the bill. Most agencies will work within their existing toolboxes, especially under time pressure.
WHAT THE EXECUTIVE BRANCH DID WELL	Organized early and conducted deep prepassage planning. Because of a year of prior conversation and technical assistance, the VA already knew many critical strategic and planning details before the bill passed. It was not starting from zero on the day of enactment. Used existing agile capacity-based contracts to move fast. By modifying existing contracts rather than initiating new procurements, the VA avoided a 12- to 16-month procurement delay and was able to stand up teams rapidly. Prioritized IT stabilization and scalability early. The VA CIO prioritized modernizing the IT infrastructure, building monitoring systems, and increasing system capacity before the surge in demand arrived. Maintained intensive engagement and transparency with Congress. The VA held regular briefings with the House Veterans' Affairs Committee and OMB throughout implementation, keeping Congress informed rather than reporting only after the fact.
WHAT THE VA DIDN'T ACCOUNT FOR	Underestimated scalability challenges. VA.gov and the claims systems struggled under the increased load, leading to deadline extensions. The scale of demand exceeded what had been planned for. Interoffice competition and fragmented data environments. Different VA offices used different tools and data sources, which created planning friction and inconsistencies across the implementation effort.

Common misconceptions

Below are some common statements or beliefs related to implementation.

MISCONCEPTION	REALITY
"Congress likes the structure of the budget and doesn't want you to change it."	The structure can be changed. It just requires fighting for it. Capability-based budgeting is possible and produces better outcomes.
"If we pass the law, the states will figure out implementation."	When Congress passes a bill, that's just the start. States are figuring it out at the same time the federal government is, and budgetary concerns mean these things take time.
"Hard deadlines create accountability."	Deadlines signal urgency but can produce perverse incentives. Defining success as hitting a number rather than achieving an outcome leads to the wrong behavior.
"You can go into legacy systems and build out new functionality."	There is no good track record of bolting new requirements onto old systems.
"Budget lines tell you how much was spent and what the outcome was."	Rarely does one budget line correspond to one outcome. Development, infrastructure, and platform costs are distributed across multiple systems and funding sources.

Questions to ask

- What legacy systems does this bill depend on? What are their constraints and timelines? Have the people who run those systems been consulted?
- Before the law passes, find out how many people this will affect and what it will mean for existing system capacity. Has anyone modeled the load?
- What is the implementation timeline, and does it account for hiring, procurement, and software deployment cycles and not just policy rollout?
- Are the deadlines in this bill achievable? Are they hard deadlines, or can they be structured as cascading milestones?
- Is funding structured to match the actual timeline of implementation? Is multiyear funding possible?
- How are budget lines structured? Do they correspond to outcomes, or just to program offices and acquisitions?
- Who is responsible for shepherding implementation, and is that person close enough to the work to make real decisions?
- Are you talking to the people doing the actual work, or to intermediaries? How close are your sources to the work and to the people being served?
- What channels exist for implementers to surface problems back to the Hill? Are those channels real and used?

Glossary

Capability-based budgeting: A term coined by Solitaire Carroll to describe an approach to budgeting that groups related functions, systems, and costs together so that prioritization decisions can be made more efficiently and outcomes can be more clearly attributed to spending.

Carryover: Unspent funds from one fiscal year carried into the next. Carryovers are generally frowned upon and difficult to request, which creates pressure to obligate funds quickly instead of wisely.

Cohort analysis: An assessment of how many people a policy change will affect and what it means for system capacity, staffing, and workload. Used by the VA ahead of the PACT Act to anticipate the volume of new claims.

Continuing resolution (CR): A temporary funding measure that allows the federal government to continue operating at existing funding levels when Congress has not yet passed a new budget. CRs limit what agencies can spend and complicate long-term planning.

Multiyear funding: Appropriations that span more than one fiscal year, allowing agencies to plan and execute long-term projects without the pressure of year-end spending. Generally produces better planning behavior than annual funding.

Product Operating Model: An iterative, user-centered approach to delivering digital services, borrowed from the tech industry, in which cross-functional teams continuously test, learn, and improve. Contrasts with the traditional "waterfall" model still prevalent in most government procurement.

Tech debt: The accumulated cost of shortcuts, outdated systems, and deferred modernization work. When funding is spent on the wrong priorities, tech debt grows and still has to be addressed in future budget cycles.

Waterfall development: A linear and rigid development process in which each phase (design, build, test, deploy) must be completed before the next begins. Specifications are determined before work begins, leaving no room for learning or adjustments. Common in government software procurement and development.

Chapter 4: The Product Operating Model

This chapter draws from a talk by Ann Lewis, former Director of Technology Transformation Services at the General Services Administration. What follows is a synopsis of her talks and the discussions that followed.

Between 2003 and 2012, 94 percent of federal technology projects over $10 million failed.[4] What did they all have in common?

They followed a "project" model of operating in which success is measured by whether a predetermined set of requirements was delivered on time and on budget – not by whether the product that was delivered actually works.

There is another model: the Product Operating Model. It is a tried-and-tested organizational model that has delivered superior results in both private and public institutions and at all levels of government. Below is a synopsis of what it is, why it is effective, how it works, and how to enable it.

Key messages

- **Governments largely operate under a project model.** In this model success means delivering a predefined set of requirements on time and on budget, even if desired outcomes are not achieved.
- **A product model works differently.** It starts with the desired outcome and gives implementation teams enough flexibility to work with users, test what works, and adapt requirements as they learn.
- **Product operating models are already in use.** Federal and state agencies are applying them to improve service delivery, reduce waste, and build systems that better meet public needs.
- **It's not necessary to be a technologist to use a product model.** The product model is an organizational and operational approach. Understanding it is an operational skill, not a technical one.
- **The legislative branch has an important role to play.** It starts with clearly defining the desired outcomes rather than prescribing solutions.

Project vs. product

Most large federal technology projects that fail share the same patterns of failure:

- They begin with ambitious scopes that attempt to solve many complex problems at once.
- They have extensive requirements-gathering phases but no deep user research. Instead, teams spend months documenting what they think they need without ever testing those assumptions against reality.
- They feature long contract terms with commitments to long implementation cycles that are practically guaranteed to deliver outdated solutions even if everything goes according to plan.
- Because agencies often lack the internal staff to drive these projects, they become vendor led, with contractors setting the pace and direction of the work and agency staff performing a limited oversight role.

What is the product model?

The structural differences are significant. The Product Operating Model starts from a different premise. Instead of treating a policy or a digital service as a project with a fixed specification and an end date, it treats it as a live product, one that is continuously tested, improved, and adapted based on what is actually happening in the field. It requires persistent, cross-functional teams with the authority to act; success measured by outcomes;

4. Aratani, Lori, "Most Infrastructure Projects Are Late, Over Budget. He Hopes to Fix That," *The Washington Post*, April 28, 2023.

continuous, user-centered delivery through iterative development; and ongoing funding.

Where the project model funds a project until it ships, the product model funds ongoing work because the work is never really done.

This is not a new or radical idea. The best software organizations in the world have operated this way for decades. But it's not just software organizations. Startups work cross-functionally by necessity, and some of the most effective pockets of the military operate this way.

What would it take to make this inherently nimble, cross-functional, and outcome-oriented model a more common practice in government?

What conditions need to be in place for the product model to work?

The product model does not operate in a vacuum. It requires alignment across and integration among several functions that governments tend to keep separate, including funding mechanisms, team structure and roles, roadmapping and prioritization, decision-making authority, success metrics, and oversight. When these are out of sync, even well-intentioned teams revert to the project model by default.

In the executive branch, this means continued capability-based budgeting, cross-functional delivery teams that are empowered to work iteratively, and leadership that champions outcomes over process compliance. The basic operating principle is simple: start small, center users, work iteratively, deliver regularly. Many agencies already have the cross-functional teams to do this; but they are not always structured or empowered to work that way.

	PROJECT MODEL	**PRODUCT MODEL**
Scope	Fixed	Flexible
Time	Flexible	Fixed
Cost	Pay for deliverables	Pay for outcomes
Team	Siloed	Cross-functional
Approach	Plan it all → deliver it all → see if it worked	Plan a piece → deliver a piece → see if it worked → repeat
Success	Was it delivered on time and on budget?	Did it achieve or enable the outcome we intended?

Contractors benefit enormously from the project model because it has a fixed scope, but a flexible timeline and budget. But the product model isn't anti-contractor; it requires internal capacity that enables institutions to work more effectively with contractors to achieve the desired outcome.

The role of the legislative branch here is to enable and incentivize this kind of work by, for instance:

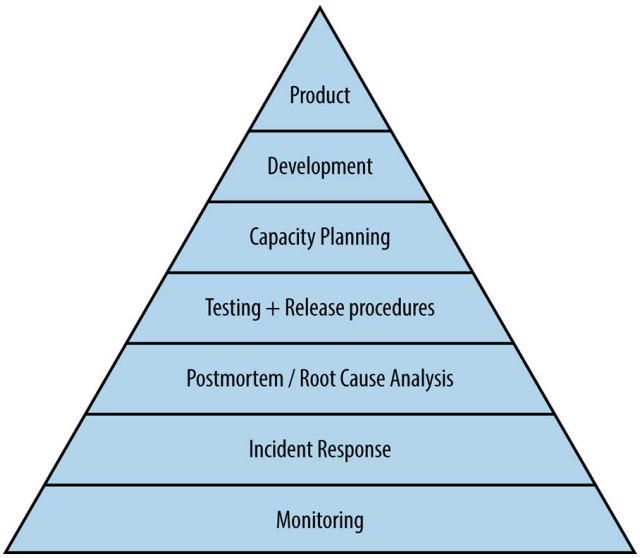
- clearly defining problems and constraints
- focusing on desired outcomes instead of features
- encouraging pilots and iteration rather than locking in a single approach from the start
- defining success through evidence and user impact, and building in the mechanisms to measure both

Staying close to the ground

Distance is one of the most common failure modes in large government-technology programs. As a program scales, the people making decisions move further from the systems and the users those systems serve. News travels up through layers of reporting such that by the time it reaches decision-makers, it has been filtered, summarized, and often sanitized.

The product model pushes against this. It requires people on the ground to be continuously assessing the systems, not just when something goes wrong but as a matter of routine. This means continuous root-cause analyses and postmortems conducted by empowered team members closest to the work rather than by the highest-ranking person or contractors.

The foundations of a good product — the bottom four layers in the hierarchy below — are ultimately about tracking outcomes and impact. This requires being close to the work.



Service Reliability Hierarchy, Google SRE book, Part III

Monitoring what is happening, the ability to respond to issues, the ability to analyze and learn from issues, and to continuously test and release pieces of the product all require being as close as possible to what is happening in practice.

Common misconceptions

Below are some common statements or beliefs related to product delivery.

MISCONCEPTION	REALITY
"You need to be a technologist to use this model."	You don't. The product model is an organizational and operational approach. Understanding it is an operational skill, not a technical one.
"If we define the requirements clearly enough upfront, the project will succeed."	Users' needs are not fully knowable until you start building and testing. Requirements change. Extensive upfront specification is one of the most reliable predictors of outdated delivery or outright failure.
"On time and on budget means success."	All it means is that the project was delivered. Whether it works for the people it was meant to serve is a separate, and fundamental, question.
"Agencies just need more money to fix their technology."	Money spent through the wrong model produces the wrong outcomes. Funding structure matters as much as funding level.
"Vendors can lead this if agencies don't have the capacity."	Vendor-led processes produce vendor-centered outcomes. Agencies need internal capacity and authority to set direction with vendors, or they lose control of the work.

Questions to ask

- Does this legislation define a problem to solve, or does it prescribe a specific solution?
- How is success defined — by delivery milestones, or by outcomes for users?
- Is the desired outcome clear?
- Does the funding structure support ongoing, iterative work, or does it fund a project until it ships and then stops?
- Are the teams responsible for implementation empowered to make decisions, or do they have to escalate every decision?
- What mechanisms exist for people close to the work to surface problems? Is there safety in doing so?
- Does this legislation encourage pilots and iteration, or does it lock in a single approach from the start?

Glossary

Cross-functional delivery team: A team that includes the full range of skills needed to deliver a product — engineering, design, policy, and operations — working together rather than in sequence.

Iterative development: A development approach in which work is broken into discrete cycles of building, testing, and improving rather than a single, extended cycle of design, build, and ship.

Product Operating Model (POM): An iterative, user-centered approach to delivering digital services borrowed from the tech industry in which cross-functional teams continuously test, learn, and improve. Contrasts with the traditional "waterfall" model still prevalent in most government procurement.

Project Model: The default operating model of most government technology programs. Characterized by rigid scopes, time-limited work, and success metrics tied to on-time, on-budget delivery rather than user outcomes.

Postmortem: A review conducted after an incident or failure to understand what happened, why, and what should change. Most useful when conducted in a culture of honesty rather than blame.

Root Cause Analysis: A process for identifying the underlying cause of a problem rather than addressing its symptoms. This is most effective when conducted by empowered team members close to the work, and when leadership is prepared to confront a reality that may not align with their assumptions.

Chapter 5: AI in implementation

This chapter draws from a discussion with Cori Zarek, former Deputy Administrator of the U.S. Digital Service, and Marianne Bellotti, author of Kill It With Fire: Manage Aging Computer Systems (and Future Proof Modern Ones). Unlike other chapters, because this was so discussion-heavy, what follows is a summary of key points, themes, and insights.

The way we think about AI in implementation should be the same as how we would think about any new technology in implementation:

- What is our goal?
- What can this do?
- Can it be useful to our goals? If so, where?
- What are indicators of success? Failure?

There is vast variability in utility, consistency, and reliability of products that fall into the very broad category of "AI." This, combined with the rapidly changing nature of the technology, makes it simultaneously critical and difficult to be clear-eyed and discerning about adoption.

Key messages

- **AI is most useful to people who already know the subject.** Users with existing expertise can frame effective prompts and spot errors, while those without it risk a false sense of confidence, and the time spent verifying output can cancel out efficiency gains.
- **Ask what is lost before automating a task.** Some tasks, like drafting memos, are how junior staff build knowledge, judgment, and expertise, so automating them can quietly erode the institution's capacity even when it saves time.
- **The most valuable uses are often the least glamorous.** Tools that track legislation, map workflows, or sort through large volumes of text (like Maryland's Legi-Assist) reduce volume so that staff can apply their judgment where it matters most.
- **Newer is not necessarily better.** Wholesale replacement of legacy systems is rarely the right answer. Stability and reliability in old systems should not be dismissed just because something newer exists. The question should not be "is this system old?" but "what problem are we actually trying to solve, and is replacement the best way to solve it?"
- **The idea of "digital sovereignty" invites interrogation.** By definition, "sovereignty" means that a government has ultimate authority to govern its territory and people without outside interference. Can you maintain sovereignty while outsourcing the systems you rely on to an external entity?
- **Security cannot be an afterthought:** prompt injection — that is, a security vulnerability in which malicious instructions are embedded into an AI system's inputs to manipulate its behavior — and other AI-specific vulnerabilities carry heightened risk in government contexts.

How fellows use and perceive AI

The fellows from both pilot cohorts reflected the breadth of the legislative branch: bipartisan cohorts representing both chambers, member offices, committees, varying levels of seniority, and a wide range of policy expertise. What follows is a synopsis of a discussion about how they perceive and use AI in their work.

The honest picture from the room was mixed. There was a huge range of use. About half had used AI in some way, and the other half had not. Of those who had used AI, about half said they found it consistently useful.

There was also a range in what AI was used for, and in the interest from office staff and leadership in adoption.

One staffer described AI as useful for getting up to speed quickly on topics outside his expertise, such as cybersecurity. When pressed on how he verifies the quality of what it produces, he said he treats it the way he would treat input from an outside expert: He takes the time to check it. Another staffer pushed back, noting that the verification process often takes more work than just researching the topic yourself, which challenges the value proposition of AI as a tool for efficiency.

The extent to which AI helps depends on what it's being used for. As a starting point for building initial familiarity with a topic, it can have real value. As a substitute for more in depth work, it is far less clear.

One excellent, albeit unanswered, question: *Where does training fit in?* Struggle is central to developing expertise. AI can shortcut that process in ways that feel productive but are dangerously limiting.

Discussion topics

The following is a summary of key topics from our discussion with Cori Zarek, former Deputy Administrator of the U.S. Digital Service, and Marianne Bellotti, author of Kill It With Fire: Manage Aging Computer Systems (and Future Proof Modern Ones)

Automating tasks

Not all AI use is the same. A useful starting point is to ask when it's useful and appropriate to automate something, and when it's not.

A lot of automation already happens in daily work without anyone calling it AI. Spell check and autocorrect use language models; we just don't think of them as Artificial Intelligence or LLMs, and we've collectively decided that's an acceptable automation.

As in many workplaces, generative AI presents both opportunities and challenges. While there are use-cases where efficiency could be gained, AI products are still not reliable or consistent — two things required for building trust with a system. AI products also raise questions about institutional capacity. For example: Writing memos is an important part of how junior staffers develop knowledge, skills, critical analysis, reasoning, and expertise. If someone already has strong writing skills, using only AI to draft and then reviewing the output is a reasonable workflow. If they don't have strong writing skills, then using AI will short-circuit their development. This is not a hypothetical concern. It is already evident on the Hill.

The question to ask before automating anything is, "What do people learn by doing this task themselves, and what is lost if they forgo that step?"

Promising use cases in government

The most useful applications of AI in government are not the most glamorous ones. Mapping workflows, tracking legislation, analyzing large volumes of text — these are areas where AI has shown the most potential to reduce the burden on staff and free up capacity for higher-order work in meaningful ways. Maryland's Legi-Assist is a good example. It is an open-source toolkit that automatically tracks and analyzes bills from the Maryland General Assembly, helping civil servants keep pace with roughly 2,000 bills per session. The tool reduces volume so that judgment can be applied where it matters.

There was general agreement among fellows that generative AI is most helpful when users know enough about the subject matter to frame effective prompts and to make meaning of responses. Otherwise, there is a high risk of the Dunning-Kruger effect, the tendency of people with a low level of knowledge or expertise in an area to overestimate their ability. AI tools tend to make people feel like instant experts, creating a false sense of confidence without the skills to validate their work.

Legacy infrastructure and AI

Most legacy government systems were originally written to optimize human work. These legacy systems are broken into discrete parts, and while AI can analyze individual components, it cannot easily cross from file to file or project to project. The context for these systems is spread across different computers and environments, which makes AI-assisted modernization harder than it sounds.

Wholesale replacement of legacy systems is rarely the right answer. The more productive approach is to identify the specific problem that needs solving and start there, rather than replacing everything at once.

Stability and reliability in old systems should not be dismissed just because something newer exists. The question should not be "is this system old?" but "what problem are we actually trying to solve, and is replacement the best way to solve it?" Newer is not necessarily better.

The sovereignty question

Dependency was a recurring theme in the discussion. Governments increasingly depend on private companies to deliver public services, and external dependencies of any kind erode sovereignty.

This issue arises in practical ways through such questions as "Who is authorized to pause or stop an AI-assisted workflow?" and "Who is monitoring the situation?" In most cases, the answer is unclear. In government contexts, that ambiguity carries serious risk given that the provider ultimately holds the practical power regardless of the terms of a contract.

Each office in government, and in some cases each country, is attempting to build its own tools to address these challenges. That approach has limits. Not every office (or every government) has the engineering capacity to build and maintain custom solutions. The question of who controls the infrastructure underneath public services is ultimately a governance question.

Security

Prompt injection — the manipulation of AI systems through malicious inputs, including in something as routine as an email — is a real and growing concern. In private organizations, it is a significant risk. In government and in organizations adjacent to government, it is even more dangerous because of the scale and scope of the potential impact. The security implications of AI-assisted workflows are not theoretical, and they are not being treated with the seriousness they deserve.

The displacement question

One fellow described being replaced by AI: His role had been to track the positions and patterns of 50 satellites. AI can now do that better than any human. His reaction was positive: It freed him to do more meaningful work within his organization. Most people, he noted, had the opposite reaction.

Additional notes

- By the end of 2025, fewer than half of public servants globally said they have taken any AI training. Capacity building is not keeping pace with adoption pressure.
- Many agencies are now required to disclose their use of AI. This is creating a dynamic in which AI becomes a version of resume enhancement at the institutional level.
- Congressional committees have shown reluctance to explore AI use, in part from a lack of capacity rather than principled opposition.
- In the legislative branch, perception weighs more heavily than in the executive branch, which makes the feedback loop around AI adoption both more important and more challenging to get right.

Common misconceptions

MISCONCEPTION	REALITY
"AI will modernize our legacy systems."	Most legacy systems were built to optimize for human work and are highly fragmented. AI can analyze components but cannot easily work across projects or systems. Replacement is rarely the right answer. Instead, identifying the specific problem and starting *there* is.
"If we mandate AI use, adoption will follow."	Even where mandates exist in the U.S., adoption is uneven. Mandates without training, capacity building, clear use cases, and senior leaders with AI discernment produce checkbox behavior, not meaningful adoption.
"Public-facing AI chatbots are a natural fit for government services."	Agencies are still grappling with the ethical and legal boundaries of these tools, particularly around sensitive information. How government communicates with constituents is not a trivial design question, and AI changes the stakes significantly.
"AI adoption is primarily a technology problem."	No, it is primarily a workforce and governance problem. Fewer than half of public servants say they have taken any AI training. Even where enthusiasm for AI is high, readiness is not.
"Older government systems are inferior and should be replaced."	Old systems can be remarkably stable and reliable. Newer is not necessarily better, and technology should not be the goal.
"Digital sovereignty is achievable through AI adoption."	Governments cannot have control over something when they depend on an external entity to provide it, and have no internal capacity to substantively engage with that entity.

Questions to Ask

- What specific problem are we trying to solve with AI, and is AI actually the best tool for it?
- Who can pause or stop this AI-assisted workflow if something goes wrong? Is that answer clear and documented?
- What do people learn by doing this task themselves, and what is lost if AI does it instead?
- What happens if the technology doesn't work? Is that an acceptable risk?
- Do people know how to use this effectively? Safely?
- For public-facing AI products, what are the ethical and legal boundaries of using AI? Who has defined them, and how are they enforced?
- Are we building on top of legacy systems or replacing them? Have we identified the specific problem we are solving, or are we replacing everything at once?
- Who controls the AI infrastructure this program depends on? What happens if that relationship changes?

Glossary

Digital sovereignty: The idea that a government or organization can maintain meaningful control over its digital infrastructure and data. In practice, this is complicated by increasing dependency on private technology providers.

Generative AI: AI that produces new content — text, code, images — in response to a prompt. Includes large language models like the ones underlying most AI assistants.

Legacy infrastructure: Older technology systems, often built to optimize for human workflows, that underpin critical government functions. These are difficult to modernize because context and logic are distributed across many systems and components.

Prompt injection: A security vulnerability in which malicious instructions are embedded in inputs to an AI system, including routine communications like emails, to manipulate its behavior.

Resume-driven development: A pattern in which technology decisions are made to enhance the credentials of the decision-maker rather than to solve an organizational problem. This is observed at both the individual and institutional levels.

Summative AI: AI that analyzes, summarizes, or synthesizes existing information. Distinct from generative AI, which produces new content. The distinction matters for how these tools are evaluated and governed.

Transpiling: Converting code written in one programming language into another. A lower-risk, more targeted form of modernization than full system replacement.

Chapter 6: Legislating for implementation

This final chapter is a synthesis of a workshop with fellows during both cohorts. It captures what fellows themselves articulated that they want to stop doing, start doing, and continue doing, based on what they learned during the program.

STOP

Writing laws with hyperspecific guidelines. Narrow proxies for goals are not the same as goals. Overly prescriptive legislation creates rigidity, invites legal challenge, and leaves agencies without the flexibility they need to deliver results. Bills with overly prescriptive recommendations shape implementation in ways that are hard to undo, and are often costly in myriad ways long-term.

Asking for reports unless they are clearly necessary. Reporting requirements feel like accountability, but often they are just busy work for agencies that are already stretched, and rarely yield much real value. Before writing a reporting requirement into a bill, ask what decision it will inform and who will act on it. Bonus: Allow agencies to stop producing reports that are no longer useful.

Treating technology and risk aversion as the antidote to government bureaucracy. The answer is not to throw technology at every problem, or eliminate guardrails. Risk is a spectrum, and where government operates on that spectrum should be a deliberate choice based on stakes, not a default setting driven by fear or by the assumption that more technology means better outcomes.

Blaming the fixer for the crisis. When Congress scrutinizes a system that is actively being modernized and uses that scrutiny primarily for sound bites, the people trying to fix the problem pay the price — and the people the system is supposed to serve pay an even bigger one. As one guest speaker noted, none of it helped the veterans who used the system. There are two tracks in oversight: theater and actual work. Learning to tell the difference is one of the most important things a legislative office can do. Hearings are now internally recognized, even by bodies like the CBO, as sources of sound bites rather than substantive information.

Eleventh-hour negotiations. They rarely work. The conditions that produce good legislation — trust, relationships, genuine engagement across the aisle — cannot be manufactured at the last minute.

Vanity metrics. Arbitrary metrics or targets can create far more problems than they solve. While the intent is to create a measurable goal that people can be held accountable to, in practice those goals warp incentives for public institutions to prioritize the metric instead of improving the intended outcome.

START

Focusing on goals over process. Define what success looks like in terms of outcomes, not outputs. Write legislation around intent, not quantitative proxies that can be gamed. Less “Did the agency file the report on time?” and more “Are veterans getting care? Why or why not?”

Being more thoughtful about AI. AI tools should be used only where they genuinely help, and after thoughtful testing. The key word is thoughtful: What is being automated and why? What happens if a mistake is made? Is this an acceptable risk? Does this institution have the internal capacity to implement and manage AI within its systems?

Giving agencies the flexibility to work together. There is no world in which every agency builds its own IT infrastructure separately, and no world in which that would even end well. If government continues to allow fiefdoms to flourish, the demand for what is ultimately duplicative IT can become a serious fiscal problem. We need to consider where economies of scale across agencies can be valuable and appropriate.

Building effective and honest feedback loops. Honest, constructive conversation between the Hill and implementing agencies is possible, but it requires creating the conditions for it. If the political environment makes everything look like an attack rather than a conversation, feedback loops will not function. The goal is to create spaces with a foundation of trust that are stronger than the default environment — spaces in which people are looking for honest conversation, not theater. The levers for this exist: empowering agencies to adopt the Product Operating Model, engaging in closed-door conversations with staff and members of Congress, and choosing substance over sound bite.

CONTINUE

Watching how programs are actually administered. Not just at launch, but over time. This is the feedback loop in practice: What is being delivered? Who is being served? What is breaking? The answers are available if you build relationships in which it is safe enough to surface them.

Investing in technical talent. The capacity to deliver on policy commitments depends largely on the people doing the work, and those people need to be hired, empowered, and given an environment that enables them to work effectively.

Setting honest expectations. Lofty goals are motivating, but when those goals are unrealistic, they may create incentives to falsify data and lead to broken systems. Help set expectations that are ambitious and achievable, and be willing to revise them when reality requires it.

Embracing errors as part of the process. Mistakes are part of any process — an effective team can both identify and respond to errors or issues quickly. A Product Operating Model allows teams to surface errors faster, by designing, testing, and delivering continuously. Allowing small errors to surface early and often allows teams to catch them before they compound into a large failure that is only visible at the end. Encourage more of this by asking for regular demonstrations of working software, and for evidence that it has been tested with real users in their day-to-day environment.

Questions to ask

- Is writing a bill the right answer here, or is there a non-legislative intervention that would be more effective?
- Does this reporting requirement inform a specific decision? Who will act on it?
- Are we defining success by outcomes or by outputs? Are we asking for demos, or written reports and arbitrary metrics?
- What is the risk profile of this program? Does the oversight approach match the stakes?
- Have we built in mechanisms for honest feedback, including closed-door briefings, direct access to implementers, and space to surface bad news without political consequences?
- Are we engaged in how this program is being administered over time and not just at launch? *How* are we engaging?

Appendix A: Key Messages

The following summarizes the core messages of each chapter.

Chapter 1 // Why Policy Fails	Policy cannot be separated from its implementation. Implementation is shaped by four interdependent components. Technology is an invisible dependency. Good legislation *enables* good implementation.
Chapter 2 // The Goldilocks Problem	There are gaps between what a law says, what it intends, and how it gets operationalized. Anything left unaddressed in the legislative process can end up in court. Legislation needs to be clear, but not necessarily specific. The people who know what's broken often cannot reach the people who need to know.
Chapter 3 // Implementation	Passing a bill is the beginning, not the end. Federalism multiplies the complexities of implementing laws. Procurement is policy. What Congress does before a bill passes matters enormously. You cannot just bolt new requirements onto legacy systems. Budget lines usually do not correspond to outcomes. Multiyear funding changes behavior. The product model works; the project model doesn't.
Chapter 4 // The Product Operating Model	Governments largely operate under a project model. A product model works differently. Product operating models are already in use. It's not necessary to be a technologist to use a product model. The legislative branch has an important role to play.
Chapter 5 // AI in Implementation	AI is most useful to people who already know the subject. Ask what is lost before automating a task. The most valuable uses are often the least glamorous. Newer is not necessarily better. The idea of "digital sovereignty" invites interrogation. Security cannot be an afterthought.
Chapter 6 // Legislating for Implementation	*Since this class was a reflection and discussion, the messages here are what fellows articulated they wanted to stop doing, start doing, and continue doing — based on the program.*

STOP

- **Writing laws with hyperspecific guidelines.** Narrow proxies for goals are not the same as goals. Overly prescriptive legislation creates rigidity, invites legal challenge, and leaves agencies without the flexibility they need to deliver results. Bills with overly prescriptive recommendations shape implementation in ways that are hard to undo, and are often costly in myriad ways long-term.
- **Asking for reports unless they are clearly necessary.** Reporting requirements feel like accountability, but often they are just busy work for agencies that are already stretched, and rarely yield much real value. Before writing a reporting requirement into a bill, ask what decision it will inform and who will act on it. Bonus: Allow agencies to stop producing reports that are no longer useful.
- **Treating technology and risk aversion as the antidote to government bureaucracy.** The answer is not to throw technology at every problem, or eliminate guardrails. Risk is a spectrum, and where government operates on that spectrum should be a deliberate choice based on stakes, not a default setting driven by fear or by the assumption that more technology means better outcomes.
- **Blaming the fixer for the crisis.** When Congress scrutinizes a system that is actively being modernized and uses that scrutiny primarily for sound bites, the people trying to fix the problem pay the price — and the people the system is supposed to serve pay an even bigger one. As one guest speaker noted, none of it helped the veterans who used the system. There are two tracks in oversight: theater and actual work. Learning to tell the difference is one of the most important things a legislative office can do. Hearings are now internally recognized, even by bodies like the CBO, as sources of sound bites rather than substantive information.
- **Eleventh-hour negotiations.** They rarely work. The conditions that produce good legislation — trust, relationships, genuine engagement across the aisle — cannot be manufactured at the last minute.
- **Vanity metrics.** Arbitrary metrics or targets can create far more problems than they solve. While the intent is to create a measurable goal that people can be held accountable to, in practice those goals warp incentives for public institutions to prioritize the metric instead of improving the intended outcome.

START

- **Focusing on goals over process.** Define what success looks like in terms of outcomes, not outputs. Write legislation around intent, not quantitative proxies that can be gamed. Less “Did the agency file the report on time?” and more “Are veterans getting care? Why or why not?”
- **Being more thoughtful about AI.** AI tools should be used only where they genuinely help, and after thoughtful testing. The key word is thoughtful: What is being automated and why? What happens if a mistake is made? Is this an acceptable risk? Does this institution have the internal capacity to implement and manage AI within its systems?
- **Giving agencies the flexibility to work together.** There is no world in which every agency builds its own IT infrastructure separately, and no world in which that would even end well. If government continues to allow fiefdoms to flourish, the demand for what is ultimately duplicative IT can become a serious fiscal problem. We need to consider where economies of scale across agencies can be valuable and appropriate.
- **Building effective and honest feedback loops.** Honest, constructive conversation between the Hill and implementing agencies is possible, but it requires creating the conditions for it. If the political environment makes everything look like an attack rather than a conversation, feedback loops will not function. The goal is to create spaces with a foundation of trust that are stronger than the default environment — spaces in which people are looking for honest conversation, not theater. The levers for this exist: empowering agencies to adopt the Product Operating Model, engaging in closed-door conversations with staff and members of Congress, and choosing substance over sound bite.

CONTINUE

- **Watching how programs are actually administered.** Not just at launch, but over time. This is the feedback loop in practice: What is being delivered? Who is being served? What is breaking? The answers are available if you build relationships in which it is safe enough to surface them.
- **Investing in technical talent.** The capacity to deliver on policy commitments depends largely on the people doing the work, and those people need to be hired, empowered, and given an environment that enables them to work effectively.
- **Setting honest expectations.** Lofty goals are motivating, but when those goals are unrealistic, they may create incentives to falsify data and lead to broken systems. Help set expectations that are ambitious and achievable, and be willing to revise them when reality requires it.
- **Embracing errors as part of the process.** Mistakes are part of any process — an effective team can both identify and respond to errors or issues quickly. A Product Operating Model allows teams to surface errors faster, by designing, testing, and delivering continuously. Allowing small errors to surface early and often allows teams to catch them before they compound into a large failure that is only visible at the end. Encourage more of this by asking for regular demonstrations of working software, and for evidence that it has been tested with real users in their day-to-day environment.

Appendix B: Key Messages

Each chapter surfaces misconceptions that are both common and consequential. They are aggregated here for quick reference.

Chapter 1: Why Policy Fails

MISCONCEPTION	REALITY
"Technology is an implementation detail, and the agency can figure that out later."	Technology is policy infrastructure. It has dependencies, timelines, and constraints that must be understood before a bill is finalized, not after. Treating it as an afterthought is one of the most reliable predictors of implementation failure.
"If we write it into the bill, agencies will do exactly what we want."	Mandating an outcome does not create the capacity to deliver it. Agencies can only do what their people, systems, and processes allow. Legislation that ignores these constraints sets implementation up for failure.
"The government cannot attract good technical talent."	The mission and scale of public service work are significant draws. Attracting talent is not the main challenge; creating environments in which talented professionals can do meaningful work is.
"This IT product should have been done months ago."	Yes, at some point the product needs to launch and be usable, but software is never really "done." This is both because there is always room for improvement based on actual use and because of how quickly the technology landscape changes. There must be room for ongoing adjustments of all kinds.

Chapter 2: The Goldilocks Problem

MISCONCEPTION	REALITY
"Writing it into the bill will force the agency to do it."	Mandating an outcome doesn't create the capacity to deliver it. Overspecification creates rigidity and, often, new failures.
"Hard deadlines create accountability."	They signal urgency but rarely produce it. They can also produce perverse incentives when the goal becomes achieving an arbitrary metric, rather than achieving the intended outcome.
"The law says what we meant."	Legislative intent, statutory text, and agency operationalization are three different things. Assume gaps among all three.
"If experts say it's a good idea, we can write it into law."	Outside experts may not understand government-specific constraints. The bill will reflect whatever knowledge was — or wasn't — in the room.

Chapter 3: Implementation

MISCONCEPTION	REALITY
“Congress likes the structure of the budget and doesn’t want you to change it.”	The structure can be changed. It just requires fighting for it. Capability-based budgeting is possible and produces better outcomes.
“If we pass the law, the states will figure out implementation.”	When Congress passes a bill, that’s just the start. States are figuring it out at the same time the federal government is, and budgetary concerns mean these things take time.
“Hard deadlines create accountability.”	Deadlines signal urgency but can produce perverse incentives. Defining success as hitting a number rather than achieving an outcome leads to the wrong behavior.
“You can go into legacy systems and build out new functionality.”	There is no good track record of bolting new requirements onto old systems.
“Budget lines tell you how much was spent and what the outcome was.”	Rarely does one budget line correspond to one outcome. Development, infrastructure, and platform costs are distributed across multiple systems and funding sources.

Chapter 4: Product Operating Model

MISCONCEPTION	REALITY
“You need to be a technologist to use this model.”	You don’t. The product model is an organizational and operational approach. Understanding it is an operational skill, not a technical one.
“If we define the requirements clearly enough upfront, the project will succeed.”	Users’ needs are not fully knowable until you start building and testing. Requirements change. Extensive upfront specification is one of the most reliable predictors of outdated delivery or outright failure.
“On time and on budget means success.”	All it means is that the project was delivered. Whether it works for the people it was meant to serve is a separate, and fundamental, question.
“Agencies just need more money to fix their technology.”	Money spent through the wrong model produces the wrong outcomes. Funding structure matters as much as funding level.
“Vendors can lead this if agencies don’t have the capacity.”	Vendor-led processes produce vendor-centered outcomes. Agencies need internal capacity and authority to set direction with vendors, or they lose control of the work.

Chapter 5: AI in Implementation

MISCONCEPTION	REALITY
"AI will modernize our legacy systems."	Most legacy systems were built to optimize for human work and are highly fragmented. AI can analyze components but cannot easily work across projects or systems. Replacement is rarely the right answer. Instead, identifying the specific problem and starting there is.
"If we mandate AI use, adoption will follow."	Even where mandates exist in the U.S., adoption is uneven. Mandates without training, capacity building, clear use cases, and senior leaders with AI discernment produce checkbox behavior, not meaningful adoption.
"Public-facing AI chatbots are a natural fit for government services."	Agencies are still grappling with the ethical and legal boundaries of these tools, particularly around sensitive information. How government communicates with constituents is not a trivial design question, and AI changes the stakes significantly.
"AI adoption is primarily a technology problem."	No, it is primarily a workforce and governance problem. Fewer than half of public servants say they have taken any AI training. Even where enthusiasm for AI is high, readiness is not.
"Older government systems are inferior and should be replaced."	Old systems can be remarkably stable and reliable. Newer is not necessarily better, and technology should not be the goal.
"Digital sovereignty is achievable through AI adoption."	Governments cannot have control over something when they depend on an external entity to provide it, and have no internal capacity to substantively engage with that entity.

Chapter 6: Legislating for Implementation

None for this chapter, because the final week was a reflection workshop aimed at actionable insights.

Appendix C: Questions to Ask

These questions are intended to guide your thinking at any stage of the legislative process. They are organized by chapter but designed to be used together.

Building the habit of asking these questions routinely can help a legislative office catch problems before they become crises.

Chapter 1: Why Policy Fails

- What technical systems does this bill depend on? Who knows those systems best? Have we found and spoken with them?
- Have we spoken with the agency staff who will implement this — not just formal legislative affairs offices, but the people closest to the work?
- What feedback loops exist between the Hill and implementing agencies for this program? How can we keep an open channel? Have we set the tone and priorities? Have we asked for demos instead of written reports and arbitrary metrics?
- Does the implementing team have the people, budget, procedures, and digital infrastructure to succeed? If not, does the bill address those items?
- What would "good implementation" of this bill look like? How would we know if we'd achieved it? How can we work with the agency to identify indicators of success and failure?

Chapter 2: The Goldilocks Problem

- What are we trying to achieve here?
- What technical systems does this bill depend on? Who knows those systems best? Have you talked with them?
- Can a computer evaluate the eligibility criteria in this bill? Think: bright-line tests. If not, how will it be implemented?
- Are the deadlines in this bill achievable? What happens if they aren't met?
- Who will be leading the implementation? Have they been involved from the beginning?
- How will we know if the law being implemented is working? What are indicators of success? What are indicators of failure?
- How can people who will implement the law surface challenges? What channels can we set up to increase feedback?
- What could go wrong? If something does go wrong, as it will, how would we know? How would we respond?

Chapter 3: Implementation

- What legacy systems does this bill depend on? What are their constraints and timelines? Have the people who run those systems been consulted?
- Before the law passes, find out how many people this will affect and what it will mean for existing system capacity. Has anyone modeled the load?
- What is the implementation timeline, and does it account for hiring, procurement, and software deployment cycles and not just policy rollout?
- Are the deadlines in this bill achievable? Are they hard deadlines, or can they be structured as cascading milestones?
- Is funding structured to match the actual timeline of implementation? Is multiyear funding possible?
- How are budget lines structured? Do they correspond to outcomes, or just to program offices and acquisitions?
- Who is responsible for shepherding implementation, and is that person close enough to the work to make real decisions?
- Are you talking to the people doing the actual work, or to intermediaries? How close are your sources to the work and to the people being served?
- What channels exist for implementers to surface problems back to the Hill? Are those channels real and used?

Chapter 4: The Product Operating Model

- Does this legislation define a problem to solve, or does it prescribe a specific solution?
- How is success defined — by delivery milestones, or by outcomes for users?
- Is the desired outcome clear?
- Does the funding structure support ongoing, iterative work, or does it fund a project until it ships and then stops?
- Are the teams responsible for implementation empowered to make decisions, or do they have to escalate every decision?
- What mechanisms exist for people close to the work to surface problems? Is there safety in doing so?
- Does this legislation encourage pilots and iteration, or does it lock in a single approach from the start?

Chapter 5: AI in Implementation

- What specific problem are we trying to solve with AI, and is AI actually the best tool for it?
- Who can pause or stop this AI-assisted workflow if something goes wrong? Is that answer clear and documented?
- What do people learn by doing this task themselves, and what is lost if AI does it instead?
- What happens if the technology doesn't work? Is that an acceptable risk?
- Do people know how to use this effectively? Safely?
- For public-facing AI products, what are the ethical and legal boundaries of using AI? Who has defined them, and how are they enforced?

- Are we building on top of legacy systems or replacing them? Have we identified the specific problem we are solving, or are we replacing everything at once?
- Who controls the AI infrastructure this program depends on? What happens if that relationship changes?

Chapter 6: Legislating for Implementation

- Is writing a bill the right answer here, or is there a non-legislative intervention that would be more effective?
- Does this reporting requirement inform a specific decision? Who will act on it?
- Are we defining success by outcomes or by outputs? Are we asking for demos, or written reports and arbitrary metrics?
- What is the risk profile of this program? Does the oversight approach match the stakes?
- Have we built in mechanisms for honest feedback, including closed-door briefings, direct access to implementers, and space to surface bad news without political consequences?
- Are we engaged in how this program is being administered over time and not just at launch? *How* are we engaging?

Appendix D: Glossary

Terms are defined in the chapter where they first appear and collected here as a unified reference. Some terms appear in multiple chapters.

Capability-based budgeting: A term coined by Solitaire Carroll to describe an approach to budgeting that groups related functions, systems, and costs together so that prioritization decisions can be made more efficiently and outcomes can be more clearly attributed to spending.

Carryover: Unspent funds from one fiscal year carried into the next. Carryovers are generally frowned upon and difficult to request, which creates pressure to obligate funds quickly instead of wisely.

Cohort analysis: An assessment of how many people a policy change will affect and what it means for system capacity, staffing, and workload. Used by the VA ahead of the PACT Act to anticipate the volume of new claims.

Continuing resolution (CR): A temporary funding measure that allows the federal government to continue operating at existing funding levels when Congress has not yet passed a new budget. CRs limit what agencies can spend and complicate long-term planning.

Conway's Law: A principle from software engineering stating that systems tend to mirror the communication structures of the organizations that build them. In government, fragmented agencies produce fragmented technology.

Cross-functional delivery team: A team that includes the full range of skills needed to deliver a product — engineering, design, policy, and operations — working together rather than in sequence.

Digital sovereignty: The idea that a government or organization can maintain meaningful control over its digital infrastructure and data. In practice, this is complicated by increasing dependency on private technology providers.

Generative AI: AI that produces new content — text, code, images — in response to a prompt. Includes large language models like the ones underlying most AI assistants.

Iterative development: A development approach in which work is broken into discrete cycles of building, testing, and improving rather than a single, extended cycle of design, build, and ship.

Legacy infrastructure: Older technology systems, often built to optimize for human workflows, that underpin critical government functions. These are difficult to modernize because context and logic are distributed across many systems and components.

Legislative Anxiety Cycle: A downward cycle in which fear and anxiety lead to legislating specific instructions – more mandates, process, and procedure. This over-specification backfires, and weakens agencies' sense of responsibility and accountability to deliver outcomes.

Legislative intent: What lawmakers meant when they wrote a bill. Often differs from what the statutory text actually says and from how the executive branch operationalizes it, creating multiple points of potential failure.

Multiyear funding: Appropriations that span more than one fiscal year, allowing agencies to plan and execute long-term projects without the pressure of year-end spending. Generally produces better planning behavior than annual funding.

Oversight: Congressional monitoring of executive branch implementation. Can be hearings, letters, reporting requirements, or informal communication. Most effective when continuous and collaborative rather than reactive.

Postmortem: A review conducted after an incident or failure to understand what happened, why, and what should change. Most useful when conducted in a culture of honesty rather than blame.

Procurement: The government process for contracting with external vendors for goods and services. Federal procurement rules are governed by the Federal Acquisition Regulation and tend to favor large, established vendors and specifications determined before any work is actually done.

Product Operating Model (POM): An iterative, user-centered approach to delivering digital services borrowed from the tech industry in which cross-functional teams continuously test, learn, and improve. Contrasts with the traditional "waterfall" model still prevalent in most government procurement.

Project Model: The default operating model of most government technology programs. Characterized by rigid scopes, time-limited work, and success metrics tied to on-time, on-budget delivery rather than user outcomes.

Progressive deadlines: A sequenced series of achievable milestones that compound over time, in place of a single hard deadline. Designed to create momentum and measurable progress by creating visible cumulative outcomes, rather than a single binary pass/fail at an arbitrary date.

Prompt injection: A security vulnerability in which malicious instructions are embedded in inputs to an AI system, including routine communications like emails, to manipulate its behavior.

Resume-driven development: A pattern in which technology decisions are made to enhance the credentials of the decision-maker rather than to solve an organizational problem. This is observed at both the individual and institutional levels.

Root Cause Analysis: A process for identifying the underlying cause of a problem rather than addressing its symptoms. This is most effective when conducted by empowered team members close to the work, and when leadership is prepared to confront a reality that may not align with their assumptions.

Summative AI: AI that analyzes, summarizes, or synthesizes existing information. Distinct from generative AI, which produces new content. The distinction matters for how these tools are evaluated and governed.

Tech debt: The accumulated cost of shortcuts, outdated systems, and deferred modernization work. When funding is spent on the wrong priorities, tech debt grows and still has to be addressed in future budget cycles.

Transpiling: Converting code written in one programming language into another. A lower-risk, more targeted form of modernization than full system replacement.

Waterfall development: A linear and rigid development process in which each phase (design, build, test, deploy) must be completed before the next begins. Specifications are determined before work begins, leaving no room for learning or adjustments. Common in government software procurement and development.

Appendix E: Reading List

These are recommended readings for anyone who would like to go deeper on any of the topics we covered during the program.

State Capacity and Implementation

How to Actually Implement a Policy — Santi Ruiz, *Statecraft*

The How We Need Now: A Capacity Agenda for 2025 and Beyond — Jennifer Pahlka and Andrew Greenway, *Niskanen Center*

Legislative Guide: Crafting Laws to Improve Effective Implementation — Amanda Patarino, *Niskanen Center*

Reforming Like It's 1995 — Jennifer Pahlka, *Eating Policy*

Success & Failure in Federal Service Delivery — *Service Design Collective*

How to Peel a Banana (a story about the Cascade of Rigidity) — Marina Nitze, *Eating Policy*

The Product Operating Model

Built for the Veteran: How the Product Operating Model Transformed the Department of Veterans Affairs — Solitaire Carroll, Holly Sydnor, Sara Thomas, and Victoria Hernandez-Corretjer, *Niskanen Center*

The Digital Service Was Never the Point — Jennifer Pahlka and Kelly Taylor, *Eating Policy*

The Product Operating Model: How Government Should Deliver Digital Services — Ann Lewis and Jennifer Pahlka, *Niskanen Center*

Budgeting and Procurement

Capability-Based Budgeting — Solitaire Carroll, *Niskanen Center*

How the FBI Blew $1 Billion on an Efficiency Project — Mark Lerner, *Niskanen Center*

How to Save a Billion Dollars — Ann Lewis, *Niskanen Center*

Why Federal IT Fails—And How Budgeting by Capability Can Fix It — Solitaire Carroll, *Niskanen Center*

Hiring and Talent

Beyond the Checkbox: Unlocking Skills-Based Hiring under the Chance to Compete Act — Cassandra Madison and Gabe Menchaca, *Niskanen Center*

What on Earth Is SME-QA and Why Should You Care about It? — Jennifer Pahlka, *Medium*

FAFSA - Lessons

Fixing FAFSA: Lessons from the Rollout — *dotEDU* podcast with Jeremy Singer, *American Council on Education*

GAO Gets Schooled by the Department of Education — Jennifer Pahlka, *Eating Policy*

When FAFSA Broke, They Called This Guy — Santi Ruiz, *Statecraft*

Artificial Intelligence

AI in Government: From Tools to Transformation — Ann Lewis, *Niskanen Center*

Decision Guide for Considering AI Use — *TechTonic Justice*

Insights from Building the Government AI Navigator — *Apolitical*

State Legislative Capacity in the AI Era — *POPVOX Foundation*

Systems and Unintended Consequences

The American Elevator Explains Why Housing Costs Have Skyrocketed — Stephen Smith, *New York Times*

A Fine Is a Price — Uri Gneezy and Aldo Rustichini, *Journal of Legal Studies*

The Four Pests Campaign and China's Great Famine — *VoxDev*

How Complex Systems Fail — Richard I. Cook

Conway's Law at Government Scale — Ann Lewis, *Niskanen Center*